IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of:	Confirmation No.: 4275
Namit JAIN	Examiner: RADTKE, Mark A.
Serial No.: 10/648,600	Group Art Unit No.: 2165

Filed on: August 25, 2003

For: DIRECT LOADING OF
       SEMISTRUCTURED DATA

<u>CERTIFICATE OF TRANSMISSION VIA EFS-WEB</u>

Pursuant to 37 C.F.R. 1.8(a)(1)(ii), I hereby certify that this correspondence is being transmitted to the United States Patent & Trademark Office via the Office electronic filing system in accordance with 37 C.F.R. §§1.6(1)(4) and 1.8(a)(1)(i)(C) on the date indicated below and before 9:00 PM PST.

Submission date: October 9, 2007   by _____ /ChristianANicholes#50266/_____

MS Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REPLY BRIEF UNDER 37 CFR § 1.193(b)(1)

Sir:

     This is in response to the Examiner's Answer mailed August 9, 2007. The

shortened statutory period runs until October 9, 2007.

REMARKS

1.   In the Examiner's Answer, in connection with the rejection of Claim 1, the Examiner essentially alleges that the specification defines a "hidden column" as being anything that stores values that are not displayed to a user when the database table that contains the hidden column is queried. Although it is true that hidden columns possess this property, the Examiner mischaracterizes the statement in paragraph [0040] of the specification as a "definition" of what a hidden column is. Expressing a property of a hidden column is not the same as providing a definition of a hidden column. The specification does not say that everything has this property is, by definition, a hidden column.

If the Examiner means to take every one of the specification's utterances about hidden columns to be definitions for what a hidden column is, then the Appellants are left to wonder why the Examiner has ignored the following statement in paragraph [0052] of the specification: "Hidden columns are useful for storing information that is needed for re-creating an original document from which values were obtained, especially when that information cannot be derived from the values themselves." Although this expresses another characteristic of hidden columns, this is actually no more a definition of what a hidden column is than the text from paragraph [0040] that the Examiner quotes. On the other hand, if the statement in paragraph [0040] is taken as a definition of what a hidden column is, then there is no reason why the statement in paragraph [0052] should not also be taken as a definition of what a hidden column is. No doubt, the Examiner chooses to focus on the statement in paragraph [0040] and ignore the statement in paragraph [0052] because, if the statement in paragraph [0052] is taken to define what a "hidden column" is, then the portion of Skinner that the Examiner cites as disclosing a "hidden column" will fail to satisfy that "definition." The Examiner is selectively ignoring parts of the

specification in order to produce an alleged definition that is most convenient for the Examiner.

Regardless of whether the statement in paragraph [0040] defines what a "hidden column" is, the Examiner's use of this statement to allege that Skinner discloses a "hidden column" is based on a logical error. The Examiner's logic goes something as follows: (A) paragraph [0040] says that hidden columns store values that are not displayed to a user. (B) Skinner discloses (allegedly) something that stores values that are not displayed to a user. Therefore, (C) Skinner discloses hidden columns. When scrutinized closely, the same logic could be applied in the following manner: (A) Frogs jump. (B) People jump. Therefore, (C) people are frogs. Clearly, this is an absurd conclusion to draw, and one that does not logically follow from the premises. The Examiner's conclusion similarly fails to follow logically from the premises.

In summary, although paragraph [0040] states a property of "hidden columns," paragraph [0040] does not define what a "hidden column" is (at least, not any more than paragraph [0052]—which the Examiner conveniently ignores—defines what a "hidden column" is). The Examiner's conclusion that anything that possesses this property must therefore also be a hidden column is logically flawed and incorrect. Those of ordinary skill in the art of database technology know well what a hidden column is in the context of database systems (for example, those of skill recognize that ROWIDs are values that traditionally have been stored in hidden columns). The aspect of Skinner that the Examiner calls a hidden column is not a hidden column, despite the fact that Skinner's aspect might store values that are not displayed to a user. Many different things can store values that are not displayed to a user. Many of these things are, nevertheless, not hidden columns.

2.   In the Examiner's Answer, in connection with the rejection of Claim 1, the Examiner alleges that since access restrictions are enforced (according to Skinner), private data will not be presented to the user. However, this is incorrect. The Applicants have already explained that the privacy of a class and its elements only bears on whether or not certain other classes will be able to inherit from that class. Whether or not data maintained by a private class will be presented to a user simply cannot be determined exclusively from the private nature of the class or its elements. Some private classes might very well contain methods (or functions) that cause data to be presented to a user when those methods are invoked. The privacy of a class or its elements has nothing to do with whether that class' data will be presented to a user. Although Skinner might say that "access restrictions are enforced," it does not logically follow that "private data will not be presented to the user."

Additionally, even if "hiding columns at a database level" would be indistinguishable from Skinner's approach "from the user's point of view," as the Examiner alleges, this would clearly not be the case when the "user" is a programmer who designs the program that causes data to be hidden. Apparently, the Examiner takes the position that, so long as there exists some user somewhere to whom some value is not presented, that value must be stored in a hidden column. Under this reasoning, virtually any value could be said to be stored in a hidden column. This is an absurd conclusion, however, since those of ordinary skill in the art of database systems clearly recognize that some values may be stored in hidden columns, and other values may be stored in columns that are not hidden columns (and yet other values might be stored outside of database columns altogether). Some users might not be able to access the values that are stored in non-hidden columns, but that doesn't make the non-hidden columns hidden columns.

Presumably, even if Skinner "implements a permissions model for determining access permissions and change permissions for different clients or users," as the Examiner alleges, it is likely that at least some users will be authorized, under the permissions model, to access the data to which the model applies. Thus, it would appear that the Examiner is saying that a hidden column might be a hidden column relative to some users, but a hidden column might not be a hidden column relative to some other users. This relativistic definition of a hidden column doesn't make much sense. The Appellants propose that those of ordinary skill in the art of database systems recognize that a hidden column is either a hidden column, or it isn't—regardless of the perspective of any particular user.

The Appellants presume that the Examiner would not venture to argue that Skinner's permissions model ever prevents **all** users from accessing data to which the model applies. At least, Skinner never teaches or suggests that the permissions model ever restricts **all** users from viewing a particular data item. If even one user could access such data (and there seems to be little point in storing data that cannot be accessed by anyone at all), then, even under the Examiner's odd definition, that data would not be stored in a "hidden column," for there would be at least one user to whom the data would be visible.

3.   In the Examiner's Answer, in connection with the rejection of Claim 12, the Examiner essentially analogizes the word "indicates," as recited in Claim 12, to the phrase "inherits from." If this analogy is followed to its logical conclusion, then Skinner cannot disclose the method of Claim 12 if Skinner does not disclose all of the following: (A) receiving data that conforms to a first class that inherits from two or more second classes, one of which also inherits from two or more third classes; (B) creating and populating a first data structure whose elements correspond to the two or more second

classes; and (C) creating and populating a second data structure whose elements correspond to the two or more third classes.

Even if Skinner discloses that a class can inherit from multiple classes (and Skinner does not appear to disclose this, expressly), which themselves can inherit from multiple other classes, there is no part of Skinner that describes the creation and population of a single first data structure whose elements correspond to **multiple classes from which a particular class inherits**. Even if Skinner described such a thing, Skinner would still fail to describe, further, the creation and population of a single second data structure whose elements correspond **to multiple classes from which a class that corresponds to an element in the first data structure inherits**. Skinner is never this specific. Thus, even if Claim 12 is read broadly enough to read "indicates" as "inherits from," Skinner still does not disclose, teach, or suggest the method of Claim 12.

4.   The Examiner's Answer apparently does not even attempt to refute the Appellants' previous arguments pertaining to the inappropriateness of the rejection of Claim 5.

5.   In the Examiner's Answer, in connection with the rejection of Claim 8, the Examiner points out that Skinner explicitly states that metadata (the alleged "data") may be "applied" to the table generation process. However, it does not follow that the metadata are ever loaded into a database. It is easily conceivable that Skinner's metadata might be used to generate a table without actually storing that metadata in the database.

Despite what the Examiner's Answer alleges, Claim 8 recites "wherein said generating and said **writing** are performed without causing a Structured Query Language (SQL) engine to load said data." As the Appellants discussed previously, the "writing" to which Claim 8 refers is "writing said data into one or more data blocks **in said database**" as recited in Claim 1. Therefore, if Skinner doesn't disclose that its metadata (the alleged

"data") is written into a database without causing a SQL engine to load that metadata, then the Examiner's rejection of Claim 8 is improper. Skinner apparently doesn't even disclose that its metadata (the alleged "data") is written into a database. Therefore, the Examiner's rejection of Claim 8 is improper for being based on something that Skinner doesn't actually disclose.

6.   In the Examiner's Answer, in connection with the rejection of Claim 9, the Examiner notes that Skinner discloses a "myMethods" vector that describes methods of a current class. The Examiner also alleges that a function must be looked up in order to be called. Even accepting these statements as true, it does not follow that the "myMethods" vector contains addresses for the methods that "myMethods" describes, even if addresses for those methods must be looked up in order to be invoked. Skinner does not disclose that the method descriptions in the "myMethods" vector contain addresses for the methods described. The alleged necessity of looking up such addresses prior to method invocation does not imply that the "myMethods" vector must contain such addresses.

7.   In the Examiner's Answer, in connection with the rejection of Claim 13, the Examiner alleges that the features of Claim 13 must be a part of Skinner if Skinner were to implement multiple inheritance. The Appellants respond that (A) Skinner does not appear to disclose multiple inheritance or any implementation that would permit multiple inheritance, expressly, and (B) the Examiner's summary conclusion, that this would be the only way for Skinner to implement multiple inheritance, seems to lack any factual support, and is merely a bare assertion. The Appellants respectfully submit that Skinner's approach never contemplated multiple inheritance, and that one of ordinary skill in the art at the time of the filing of the present application would not have found an obvious way to modify Skinner to implement multiple inheritance without undue experimentation. Besides, Claim 13 is rejected under 35 U.S.C. § 102, not under 35 U.S.C. § 103, so the

manner in which Skinner's teachings hypothetically might be modified to implement multiple inheritance is not even a proper consideration.

The Examiner alleges that Skinner's col. 19, lines 51-60, disclose the features of Claim 13. Apparently, although this portion of Skinner alludes to a way of mapping a child class to a parent class via "mySuperClass," this is not what Claim 13 is actually directed to. Claim 13 actually recites that the **elements** are associated with the set identifier. Reference to Claim 12 reveals that these **elements** correspond to **attributes** of a **type definition**, and not to a **type definition** itself. The Examiner apparently analogizes a "type definition" to a "class," but a mapping between a parent class and a child class is not the same as a mapping between a class and the attributes of a class. Furthermore, the simple mapping of a child class to a parent class, as Skinner allegedly discloses, would not require the population of a **plurality** of elements with a set identifier as disclosed in Claim 13.

For at least the above reasons, and those set forth in the Appeal Brief previously filed, Appellants respectfully submit that the imposed rejections are **not** viable, and respectfully solicit the Honorable Board to **reverse** each of the imposed rejections.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Dated: <u>October 9, 2007</u>   <u>/ChristianANicholes#50266/</u>
          Christian A. Nicholes
          Reg. No. 50,266

2055 Gateway Place, Suite 550
San Jose, CA 95110
(408) 414-1080, ext. 224